

PATENT APPLICATION

IMPROVED DIGITAL CAMERA DEVICE AND METHODOLOGY FOR WIRELESS PROGRESSIVE TRANSMISSION OF DIGITAL IMAGES

Inventors: VENKAT V. EASWAR, a citizen of India residing in Cupertino, California; and ERIC O. BODNAR, a citizen of The United States residing in Santa Cruz, California.

Assignee: LightSurf Technologies, Inc.

John A. Smart
Reg. No. 34,929

708 Blossom Hill Rd., #201
Los Gatos, CA 95032
(408) 395-8819; (408) 490-2853 FAX

IMPROVED DIGITAL CAMERA DEVICE AND METHODOLOGY FOR WIRELESS PROGRESSIVE TRANSMISSION OF DIGITAL IMAGES

5

RELATED APPLICATIONS

The present application claims the benefit of priority from, and is a continuation-in-part application of, commonly-owned U.S. application serial no. 09/434,703, filed November 5, 1999. Additionally, the present application is related to commonly-owned U.S. application serial no. 09/489,511, filed January 21, 2000. The disclosures of each of the foregoing applications are hereby incorporated by reference in their entirety, including any appendices or attachments thereof, for all purposes.

COPYRIGHT NOTICE

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

20

BACKGROUND OF THE INVENTION

25

The present invention relates to the field of digital cameras and digital image processing and, more particularly, to designs and techniques for reducing processing requirements and therefore size of digital cameras.

Today, digital imaging, particularly in the form of digital cameras, is a prevalent reality that affords a new way to capture photos using a solid-state image sensor instead of traditional film. A digital camera functions by recording incoming light on some sort of sensing mechanisms and then processes that information (basically, through analog-to-digital conversion) to create a memory image of the target picture. A digital camera's biggest advantage is that it creates images digitally thus making it easy to transfer images between all kinds of devices and applications. For instance, one can easily insert

5 digital images into word processing documents, send them by e-mail to friends, or post them on a Web site where anyone in the world can see them. Additionally, one can use photo-editing software to manipulate digital images to improve or alter them. For example, one can crop them, remove red-eye, change colors or contrast, and even add and delete elements. Digital cameras also provide immediate access to one's images, thus avoiding the hassle and delay of film processing. All told, digital photography is becoming increasingly popular because of the flexibility it gives the user when he or she wants to use or distribute an image.

10 The defining difference between digital cameras and those of the film variety is the medium used to record the image. While a conventional camera uses film, digital cameras use an array of digital image sensors. When the shutter opens, rather than exposing film, the digital camera collects light on an image sensor, a solid state electronic device. The image sensor contains a grid of tiny photosites that convert light shining on them to electrical charges. The image sensor may be of the charged-coupled device (CCD) or complementary metal-oxide semiconductor (CMOS) varieties. Most digital cameras employ charge-coupled device (CCD) image sensors, but newer cameras are using image sensors of the complimentary metal-oxide semiconductor (CMOS) variety. Also referred to by the acronym CIS (for CMOS image sensors), this newer type of sensor is less expensive than its CCD counterpart and requires less power.

15 During camera operation, an image is focused through the camera lens so that it will fall on the image sensor. Depending on a given image, varying amounts of light hit each photosite, resulting in varying amounts of electrical charge at the photosites. These charges can then be measured and converted into digital information that indicates how much light hit each site which, in turn, can be used to recreate the image. When the exposure is completed, the sensor is much like a checkerboard, with different numbers of checkers (electrons) piled on each square (photosite). When the image is read off of the sensor, the stored electrons are converted to a series of analog charges which are then converted to digital values by an Analog-to-Digital (A to D) converter, which indicates how much light hit each site which, in turn, can be used to recreate the image.

Early on during the digital imaging process, the picture information is not in color as the image sensors basically only capture brightness. They can only record gray-scale information -- that is, a series of increasingly darker tones ranging from pure white to pure black. Thus, the digital camera must infer certain information about the picture in order to derive the color of the image. To infer color from this black and white or grayscale image, digital cameras use color filters to separate out the different color components of the light reflected by an object. Popular color filter combinations include, for instance, a red, green, and blue (RGB) filter set and a cyan, magenta, and yellow (CMYK) filter set. Filters can be placed over individual photosites so each can capture only one of the filtered colors. For an RGB implementation, for example, one-third of the photo is captured in red light, one-third in blue, and one-third in green. In such an implementation, each pixel on the image sensor has red, green, and blue filters intermingled across the photosites in patterns designed to yield sharper images and truer colors. The patterns vary from company to company but one of the most popular is the Bayer mosaic pattern, which uses a square for four cells that include two green on one diagonal, with one red and one blue on the opposite diagonal.

Because of the color filter pattern, only one color luminosity value is captured per sensor pixel. To create a full-color image, interpolation is used. This form of interpolation uses the colors of neighboring pixels to calculate the two colors a photosite did not record. By combining these two interpolated colors with the color measured by the site directly, the original color of every pixel is calculated. This step is compute-intensive since comparisons with as many as eight neighboring pixels are required to perform this process properly. It also results in increased data per image so files get larger.

In order to generate an image of quality that is roughly comparable to a conventional photograph, a substantial amount of information must be capture and processed. For example, a low-resolution 640 x 480 image has 307,200 pixels. If each pixel uses 24 bits (3 bytes) for true color, a single image takes up about a megabyte of storage space. As the resolution increases, so does the image's file size. At a resolution of 1024 x 768, each 24-bit picture takes up 2.5 megabytes. Because of the large size of this information, digital cameras usually do not store a picture in its raw digital format but, instead, apply compression

technique to the image so that it can be stored in a standard compressed image format, such as JPEG (Joint Photographic Experts Group). Compressing images allows the user to save more images on the camera's "digital film," such as flash memory (available in a variety of specific formats) or other facsimile of film. It also allows the user to download and display those images more quickly.

During compression, data that is duplicated or which has no value is eliminated or saved in a shorter form, greatly reducing a file's size. When the image is then edited or displayed, the compression process is reversed. In digital photography, two forms of compression are used: lossless and lossy. In lossless compression (also called reversible compression), reversing the compression process produces an image having a quality that matches the original source. Although lossless compression sounds ideal, it doesn't provide much compression. Generally, compressed files are still a third the size of the original file, not small enough to make much difference in most situations. For this reason, lossless compression is used mainly where detail is extremely important as in x-rays and satellite imagery. A leading lossless compression scheme is LZW (Lempel-Ziv-Welch). This is used in GIF and TIFF files and achieves compression ratios of 50 to 90%.

Although it is possible to compress images without losing some quality, it's not practical in many cases. Therefore, all popular digital cameras use a lossy compression. Although lossy compression does not uncompress images to the same quality as the original source, the image remains visually lossless and appears normal. In many situations, such as posting images on the Web, the image degradation is not obvious. The trick is to remove data that is not obvious to the viewer. For example, if large areas of the sky are the same shade of blue, only the value for one pixel needs to be saved along with the locations of where the other identical pixels appear in the image.

The leading lossy compression scheme is JPEG (Joint Photographic Experts Group) used in JFIF files (JPEG File Interchange Format). JPEG is a lossy compression algorithm that works by converting the spatial image representation into a frequency map. A Discrete Cosine Transform (DCT) separates the high- and low-frequency information present in the image. The high frequency information is then selectively discarded, depending on the

5 quality setting. The greater the compression, the greater the degree of information loss. The scheme allows the user to select the degree of compression, with compression ratios between 10:1 and 40:1 being common. Because lossy compression affects the image, most cameras allow the user to choose between different levels of compression. This allows the user to choose between lower compression and higher image quality, or greater compression and poorer image quality.

10 Today, a variety of digital image products are available to consumers. Typically, the images recorded with these devices tend to contain a large amount of information. Regardless of how images are recorded digitally, at some later point in time, the image information must become available to other devices -- that is, become available to a larger network of digital devices, so that the images may be outputted (e.g., printed to hard copy) or shared with other people. Existing devices provide connectivity in a rather awkward fashion, for instance, requiring a user to swap the digital film (media) or make a connection (e.g., serial cable) from the imaging device to a desktop. As these approaches are tedious, at best, there is great interest in providing increased convenience to users.

15 The image quality produced by state-of-the-art digital cameras today is very good, thus making those devices suitable for portrait photography, for instance. On the other hand, such high-quality information for an image is much higher than necessary for communicating the content of that image to someone viewing it on a Web site or attached to an e-mail message, or even in a small document. At the same time, it is helpful to have that additional image information. In other words, it is desirable to not lose that information, since at a later point in time a higher quality version of that image may be desired. Consider, for instance, a user of a portable digital image device "in the field" (e.g., outdoors) who wants to communicate, as quickly and as conveniently as possible, the content of an image to some target (e.g., office), whether via a wireless or wireline connection. Today, such a user faces a dilemma. The user must choose whether to take a detailed image (i.e., recorded in high detail) or take a smaller "packaged" version of that detailed image, one more suitable for rapid transmission to a target destination (which does not need the detail right away).

20
21
22
23
24
25

“Progressive” formats of image transmission are available today for transmitting progressive degrees of image information detail. For example, the GIF (Graphics Interchange Format) file format applies a geometric or interlaced approach, such as initially rendering only certain scan lines (e.g., render only every eighth scan line), so that the user is given a crude representation of the image early on. Later, the remaining scan lines are filled in. The JPEG format, on the other hand, applies a frequency-transform approach where the image information is transposed into frequency data (e.g., 8 x 8 or 64 pixel “block”).

5 Starting from the top-left corner of a given block, each pixel in the block becomes lower and lower in priority, visually. Therefore, this image information may be transmitted in a progressive manner by sending some percentage of information from each block of the image. Later, the remaining information may be transmitted. The approach appears

10 somewhat geometric in the sense that the image starts out “blocky” and then becomes less and less mosaic as more information is transmitted.

None of these approaches have really attempted to address the above-mentioned tradeoff, though. Instead, these formats are geared towards addressing the problem of dynamic viewing: providing a user with progressive rendering of an image at a target device (e.g., a client computer running browser software), for the purpose of allowing the user to watch an image construct itself. Rather than showing a blank space where the image is to appear (e.g., on a Web page), these formats show an image that is progressively rendered. Such approaches are particularly helpful in environments where bandwidth is limited (e.g., Internet connection over a dial-up account). However, those approaches do not attempt to address the quality versus size tradeoff that users face today when recording a digital image. At the same time, often users are capturing image subject matter that is unique -- the “Kodak moment” -- that may not be available again for recording. The user is therefore faced with a difficult choice: (1) choosing a low quality recording mode for capturing a smaller image package for faster delivery or upload (i.e., for time or cost reasons) or (2) choosing a high quality recording mode before the moment is gone, but one that is ill-suited for transmission.

5

10

15

20

25

As practically all users of digital image products face limited bandwidth available for transmitting or uploading digital images, what is needed is a digital imaging device which incorporates methodology allowing users to capture a moment at a high level of quality, yet allows users to transmit an image of that moment in a manner that is suited to the communication bandwidth or medium presently available. The present invention fulfills this and other needs.

SUMMARY OF THE INVENTION

A digital imaging system of the present invention implements a methodology for distributed processing and wireless transmission of digital images. The digital image system, implemented as a digital camera in the currently-preferred embodiment, includes a Sensor, a Shutter Actuator, an Image Processor, an Image (DRAM) Memory, a (Central) Processor, a Keypad and Controls, a Program Code Flash Memory, a (System) Memory, a Direct View Display, a Hot Shoe Interface, and a "Digital Film" Flash Memory. These various components communicate with one another using a bus architecture including, for instance, an Address Bus, a Data Bus, and an I/O (Input/Output) Bus.

The basic approach adopted by the present invention is to adopt techniques for reducing the amount of processing power required by a given digital camera device and for reducing the bandwidth required for transmitting image information to a target platform. Given that digital cameras exist in a highly-connected environment (e.g., one in which digital cameras usually transfer image information to other computing devices), there is an opportunity to take advantage of other processing power that is eventually going to come into contact with the images that are produced by the digital imaging device ("imager"). More particularly, there is an opportunity to defer and/or distribute the processing between the digital imager itself and the target platform that the digital imager will ultimately be connected to, either directly or indirectly. The approach of the present invention is, therefore, to decrease the actual computation that occurs at the digital imager: perform a partial computation at the digital imager device and complete the computation somewhere else -- somewhere where time and size are not an issue (relative to the imager). By "re-architecting"

the digital camera to defer resource-intensive computations, the present invention may substantially reduce the processor requirements and concomitant battery requirements for digital cameras. Further, the present invention adopts an image strategy which reduces the bandwidth requirements for transmitting images, thereby facilitating the wireless transmission of digital camera images.

A preferred methodology of the present invention for digital image processing includes the following steps. At the outset, an image is captured by a capture process; this may be done in a conventional manner. Next, however, the color interpolation or transformation process of conventional digital image processing is entirely avoided. Instead, the sensor image is separated into individual color planes (e.g., R, G, and B planes for an RGB color filter mosaic). Each color plane consists of all the sensor pixels imaged with the corresponding color filter. The color plane separation process requires far fewer machine instructions than the color interpolation and transformation process. The separated color plane information is referred to as "luminosity information". Hence as described herein, operations on the "luminosity" image refer to operations applied to the individual color planes in the luminosity image. Next, the methodology of the present invention immediately proceeds to coding the luminosity information (i.e., the separated color planes). The present invention applies a wavelet transform process to prioritize information in the luminosity image (i.e., the color planes in the luminosity image are individually wavelet-transformed). Those skilled in the art, enabled by the teachings of the present invention, will recognize that the wavelet transformation described herein could easily be replaced by other transform decompositions (e.g., Discrete Cosine Transform (DCT), such as used in JPEG) while still being compatible with the present invention.

The wavelet transform process or technique may be thought of as a process that applies a transform as a sequence of high- and low-pass filters. In operation, the transformation is applied by stepping through the individual pixels and applying the transform. This process, which creates an image that contains four quadrants, may for instance be performed as follows. First, a high-pass transform then a low-pass transform is performed in the horizontal direction. This is followed by a high-pass transform then a

low-pass transform performed in the vertical direction. The upper-left quadrant is derived from a low-pass horizontal/low-pass vertical image; the lower-left quadrant comprises a high-pass horizontal/low-pass vertical image; the upper-right quadrant comprises a low-pass horizontal/high-pass vertical image; and the lower-right quadrant comprises a high-pass horizontal/high-pass vertical image. The result of this is that the information most important to the human eye (i.e., the information, that from a luminosity or black and white perspective, the human eye is most sensitive to) is in the high-priority “low/low” quadrant, that is, the upper-left quadrant which contains the low-pass horizontal/low-pass vertical image. Most of the information in the other three quadrants, particularly the lower-right quadrant, is fundamentally zero (when based as an onset of a center frequency), that is, image information that is least perceived by the human eye. Thus, the low/low quadrant is considered the highest-priority quadrant, with the remaining quadrants being considered to be of much lower priority.

In basic operation, the transform process consists of processing the image as a whole in a stepwise, linear fashion. For instance, when processing the image in a horizontal direction, one would take a horizontal vector of image data (e.g., seven horizontal neighboring pixels) and multiply that by a predetermined set of coefficients (e.g., seven coefficients for a seven-pixel vector). This yields a single pixel value. Then the process continues in a sliding-window fashion by shifting over by some number of pixel(s) (e.g., two pixels), for processing the next vector of seven horizontal neighboring pixels. The transform process may be repeated multiple times, if desired. When repeated, the process of applying high- and low-pass filters is repeated for the low/low quadrant of the then-current image (i.e., the prior result of high-pass horizontal and vertical filtering), again generating a four-quadrant image. Those skilled in the art will recognize that the filtering process can be applied to the other quadrants (e.g., low/high, and the like) as well. Further, the filtering operations can be continued recursively, further decomposing each quadrant into four sub-quadrants and so forth and so on. These quadrants are also referred to as “bands”, in the image processing literature. Whether the image is transformed with a single pass or multiple passes, the end result is still a wavelet-transformed image, which may then be readily

compressed (e.g., using quantization, followed by entropy coding schemes like run-length encoding and Huffman coding).

After generating the wavelet-transformed image, the preferred methodology of the present invention proceeds to apply quantization to the image. This process involves dividing the wavelet-transformed data by a number (called the “quantization step size”) to reduce the bit depth of the wavelet data. The step size can be changed for each band of the wavelet data. Typically higher frequency bands are divided by larger numbers to de-emphasize the bands. Correspondingly, the wavelet data is “dequantized,” i.e., multiplied by the quantization step size during decompression (at the server/desktop). The process of quantization and dequantization involves loss of precision, and is typically the only lossy stage during compression. At this point, the image information (i.e., all quadrants and subquadrants) can be compressed as if it were fundamentally just a normal binary file. Thus, one can apply a simple, conventional compression as a compute-efficient compression process. In a preferred embodiment, the compression process is actually performed in two stages. In a first stage, run-length encoding (RLE) is applied to compress the image data. The insignificant regions of the image data (i.e., the regions that intersect high pass filters) tend to be predominantly centered around a single value; these can be compressed substantially. When applying run-length encoding to this type of information, for instance, one gets extremely long runs of similar data. Thus, in a preferred embodiment, the image data is compressed in a first stage using run-length encoding. This target result may then, in turn, be further compressed using Huffman coding, for generating a final compressed luminosity record that is suitable for storage on a digital camera and for wireless transmission.

Thus as described above, the camera-implemented portion of image processing foregoes color processing. Instead of performing compute-intensive tasks, such as color interpolations and YUV transformations (Y representing brightness or luminance, and U and V representing degree of colors -- hue and saturation), the methodology performs trivial color plane separation. This is followed by wavelet decomposition, quantization, and generic binary compression (e.g., run-length and Huffman encoding).

The end result is that the amount of processing necessary to go from a captured image to a compressed record of the captured image (i.e., a record suitable for storage on the digital camera) is substantially less than that necessary for transforming the captured image into color and then compressing it into a color-rendered compressed image. 5 Further, the resulting compressed luminosity record, because of its increased compression ratios (e.g., relative to conventional JPEG), facilitates wireless (or other limited bandwidth) transfer of images to target platforms.

A methodology of the present invention for wireless progressive transmission of digital images is also provided. The digital imaging device incorporates a progressive transmission methodology allowing a user to capture a moment at a high level of 10 quality/resolution, yet transmit a version of that image in a manner that is adapted for the communication bandwidth or medium currently available. The approach adopted by the present invention is to incorporate a software-implemented methodology (and supporting file format) into the imaging device itself that records (or is capable of recording) a moment in its full digital quality/resolution (desired by the user) but then prioritizes the image information 15 into a variety of logical layers or compartments. Using this partitioning approach, the system can optimize how much or how little image information is actually transmitted to a target destination at a given point in time, for instance, based on the then-existing communication medium available. Notwithstanding this flexibility, the user has at all times a full 20 quality/resolution recording of the image.

In typical usage, therefore, the user records an image at its full quality/resolution on the digital film of his or her imaging device, that is, utilizing all layers 25 (e.g., the full nine-layer file for a nine-layer embodiment). When uploading the image, however, the user may configure the system to transmit only the first layer, reducing the total amount of image information needed to be sent and thus reducing transmission time and bandwidth requirements. Here, the user may record an image at its full digital quality/resolution yet send only a version of that image that is cost-effective and quick to transmit. Further in accordance with the present invention, the user may elect to collaborate or synchronize the differences between the image information captured (e.g., full nine-layer

file) with the image information already transmitted to a target destination (e.g., first layer only), at the user's convenience, so that the transmitted image may be upgraded to its final, full quality/resolution representation.

GLOSSARY

5 *Cell*: Image sensors are overlaid with a color filter array that has a fixed repeating pattern. The pattern is typically 2 pixels wide and 2 pixels high, and has 2 green filters and 1 red and 1 blue filter. This repeating pattern is called a “cell”. The above pattern is a particular example that is often referred to as a “Bayer” pattern.

10 *Channel*: Channels are the axes of a color space. For example the R, G, B color space has Red, Green, and Blue channels.

15 *Color space*: This is typically a three-dimensional space that represents the color of image pixels. Image pixels are typically represented by the three triplet colors: red (R), green (G), and blue (B). R, G, and B can vary between 0.0 (minimum) and 1.0 (maximum). The image is represented by a primary channel comprising Green (G) and secondary channels comprising Red (R) and Blue (B). In addition to the RGB color space, other color spaces, like CMYK (Cyan, Magenta, Yellow), HSV (Hue, Saturation, Value), and YUV (Luminance, Chrominance U, and Chrominance V), are also used in the literature. Typically these color spaces are related by matrix transformations.

20 *Mosaic*: A mosaic is the image generated by an image sensor overlaid with a color filter array.

25 *Primary and secondary channels*: The primary channel is the color space axis that is most important for human perception. The secondary channels are the remaining two channels. For example, in the YUV color space, the Y channel is the primary channel and U and V are secondary channels. In the RGB color space, G is the primary channel and R and B are secondary channels.

30 *Sub-band coding*: The general class of image compression techniques (including the wavelet- based coding method) where an image is decomposed into various bands and the bands compressed using techniques suited to the band characteristics.

35 *Transform-based compression*: Transform-based compression involves subjecting each color plane of an image by a mathematical transform (e.g., such as the Fourier Transform, Cosine Transform, or Wavelet Transform). The transformed plane is more amenable to image compression. The transformed plane is quantized (i.e., fewer bits are retained per pixel than in the original) and compressed using entropy coding techniques, like Huffman or arithmetic coding.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1A is a block diagram illustrating a digital camera device suitable for implementing the present invention.

5 Fig. 1B is a block diagram illustrating a digital computer suitable for implementing distributed processing portions of the present invention.

Fig. 2 is a block diagram of a software system suitable for controlling the computer of Fig. 1B.

10 Fig. 3A is a block diagram illustrating a methodology of the present invention for distributed digital image processing (including contrasting it with conventional digital image processing).

Fig. 3B is a block diagram illustrating a multi-pass wavelet transform process.

Figs. 3C-I are black and white photographic images that compare the results of JPEG compression with wavelet transform.

Fig. 4A is a block diagram illustrating overall processing at a target platform (e.g., server or desktop computer).

Fig. 4B is a block diagram illustrating method steps of the present invention for completing image processing at a target platform (e.g., server or desktop computer).

Fig. 5A is a graph illustrating the interplay between quality and resolution when rendering an image.

20 Fig. 5B is a graph illustrating division of quality/resolution combinations into various layers, to form a matrix or grid.

Fig. 5C is a graph illustrating that transmission of layers should be selected so as to fall within a single bounding box.

25 Fig. 6A is a block diagram illustrating the basic data organization or progressive file format employed in a preferred embodiment of the present invention.

Fig. 6B is a flowchart illustrating the methodology of the present invention for organizing information into the progressive file format of the Fig. 6A.

Fig. 7 is a flowchart illustrating the methodology of the present invention for transferring image data to a destination device (e.g., server computer).

Fig. 8 is a block diagram illustrating a binary large object (blob) embodiment of the basic data organization or progressive file format of the present invention, for use in devices not supporting physical files.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

The following description focuses on an embodiment of the present invention in a digital camera device, which is the currently-preferred embodiment. However, those skilled in the art will appreciate that the present invention may be embodied in other image capturing/recording/processing devices, including, for instance, video phones, closed-circuit cameras, video camcorders, or other devices capable of capturing, recording, and/or processing images. Further, the description will focus on implementation of portions of the invention in an Internet-connected environment including desktop and server computers, such as an IBM-compatible computer running under Microsoft® Windows 2000. The present invention, however, is not limited to any particular one application or any particular environment. Instead, those skilled in the art will find that the system and methods of the present invention may be advantageously embodied on a variety of different platforms, including Macintosh, Linux, BeOS, Solaris, UNIX, NextStep, and the like. Therefore, the description of the exemplary embodiments which follows is for purposes of illustration and not limitation.

Basic System

A. Digital camera hardware

Fig. 1A is a block diagram illustrating a basic image capturing and recording system 100 suitable for implementing the present invention. For purposes of illustration, the following will focus on implementation of system 100 as a digital camera. However, as noted above, for purposes of implementing the methodology of the present invention, the system 100 may also be implemented in a variety of other digital image devices.

As shown in Fig. 1A, the system 100 includes a Sensor 101, a Shutter Actuator 103, an Image Processor 102, an Image (DRAM) Memory 104, a (Central) Processor 106, a Keypad and Controls 108, a Program Code Flash Memory 107, a (System) Memory 105, a Direct View Display or Viewfinder 109, a Hot Shoe Interface 110, and a “Digital Film” Flash Memory 111. As illustrated, these various components communicate

with one another using a bus architecture including, for instance, an Address Bus, a Data Bus, and an I/O (Input/Output) Bus.

The system 100 employs the Sensor 101 for basic image capture. The Sensor 101 operates, in essence, by capturing light and transforming that into electrical voltage levels. A suitable sensor is available from a variety of vendors, including VLSI Vision, Motorola, and Toshiba. In a preferred embodiment, the Sensor 101 includes, for example, a 1280 x 1024 color CMOS sensor, such as a VLSI Vision VVL 6801 CMOS sensor. However, other sensor technology is suitable, including CCD sensors.

The Sensor 101 must, of course, be part of a larger assembly to operate. Specifically, the Sensor 101 operates in conjunction with a lens assembly (not shown), or other optics to focus an image onto the sensor. The optics themselves are controllable, for instance, using a conventional aperture, focus, and shutter control mechanisms. The currently-preferred embodiment uses an 18 mm fixed-focal length, fixed-aperture lens assembly to provide a broad depth of field. The lens assembly employs two manual slide controls, a macro lens control, and an exposure control. The macro control switches from normal to close-up mode by sliding a macro lens in and out of the lens assembly to provide normal or extreme close-up capability. The exposure control switches from normal to bright light by sliding a neutral gray filter in and out of the lens assembly. Aside from choosing normal or bright light, normal or close-up mode, the camera requires no manual focusing, shutter speed or aperture adjustment. Operation is as simple as point and shoot. The Sensor 101, on the other hand, operates under control of the Image Processor 102, which will now be described.

The Image Processor 102, which basically operates as a state machine, provides overall control for the Sensor 101. In operation, the Image Processor 102 controls the Sensor 101 by, in effect, telling it what to do and when. For instance, the Image Processor 102 issues timing signals to the Sensor 101 for indicating how the Sensor 101 should record and stream out image data. Further, the Image Processor 102 provides general Input/Output (I/O) control that allows one to coordinate control of the sensor with other electromechanical peripherals, such as a shutter, lens aperture, or the like.

Actual implementation of the Image Processor 102 itself may be accomplished in a variety of different ways. For a microprocessor-based implementation, for instance, the Image Processor 102 may be implemented as a microprocessor (e.g., PowerPC 823 5 microprocessor, available from Motorola, Inc. of Schaumburg, IL) with DSP (digital signal processing) logic blocks, memory control logic blocks, video control logic blocks, and interface logic. Alternatively, the Image Processor 102 may be implemented as a "camera on a chip(set)" using, for instance, a Sierra Imaging Raptor I or II chipset (available from Sierra Imaging, Inc. of Scotts Valley, CA), a Sound Vision Clarity 1 or 2 chipset (available from Sound Vision, Inc. of Framingham, MA) or similar chipset that integrates a processing core 10 with image processing periphery. In a preferred embodiment, the Image Processor 102 preferably supports hardware implementation of a wavelet transform engine complete with a wavelet transform filter bank, so that the wavelet transform process may be pipelined through a series of dedicated hardware gates (instead of executed as a sequence of software 15 instructions repeatedly loaded and processed by a general-purpose microprocessor).

15 The Image Processor 102 is not a stand-alone part but, instead, relies on the (Central) Processor 106 for control instructions. The Image Processor 102 sits on the Address and Data Buses and is accessible by the Processor 106 through a series of registers. 20 In this manner, the Processor 106 may instruct the Image Processor 102 what to perform and when. For instance, the Processor 106 may instruct the Image Processor 102 to turn on the Sensor 101, to capture an image at the Sensor 101, and to execute the wavelet transform. Therefore, the Image Processor 102 is very much a facilitator but is not in and of itself a controller for the system.

25 The Shutter Actuator 103 is a simple, generic component for controlling light exposure on the Sensor 101. Depending on the behavior of the actual sensor employed, the Shutter Actuator 103 may not even be necessary. In particular, the Shutter Actuator 103 is employed in those instances where the Sensor 101 requires a black reference. In such an embodiment, the Shutter Actuator 103 is an electromechanical interface coupled to a solenoid which, when the interface responds to a particular logic level, triggers an open/close cycle of a mechanical shutter. The mechanical shutter, which serves to selectively block light

entering the lens assembly of the camera, may be of a conventional design available from a variety of suppliers. A suitable supplier includes, for instance, Sunex, Inc. of Carlsbad, CA.

The Image Memory (DRAM) 104 serves to store the image captured from the sensor. The Sensor 101 itself does not "store" the image that it captures. Therefore, the Image Memory 104 is an image capture and in-place transform (frame) buffer. This memory is controlled by the Image Processor 102 and can be shut off when not in use for power saving purposes. During basic operation of the camera, the captured image is transferred directly into the Image Memory 104, using a sample/transfer technique. In order to make this efficient, the process is controlled by the Image Processor 102 in a manner somewhat akin to DMA (direct memory access) transfer employed on desktop computers. Here, the Image Processor 102 functions as a state machine which simply samples and transfers information from the Sensor 101 to the Image Memory 104. In the presently-preferred embodiment, the Image Memory 104 comprises conventional DRAM (dynamic random-access memory) memory available from a variety of vendors, including, for instance, Toshiba, Micron, Hitachi, Samsung, and others. A size of about 4 MB (megabyte) or more is suitable for this component.

The next several components discussed, which may be viewed as components hanging off of the Address and Data Buses of the Processor 106, are typical components that one would ordinarily expect to find when implementing a data processing device; collectively, these components may be viewed as a computer embedded in the camera. For example, these components include the previously-mentioned general-purpose microprocessor (Processor 106) coupled to memory (System Memory 105 and Program Code Flash Memory 107). The Working or System Memory 105 is the general working or scratchpad memory for the Processor 106. This memory is used for storing program-created variables, stacks, heap(s), and the like. In the presently-preferred embodiment, the System Memory 105 comprises static RAM (e.g., SRAM), which is also available from a variety of vendors. A size of about 128 KB (kilobyte) or more is suitable for this purpose. The Program Code Flash Memory 107, on the other hand, comprises 1 MB of directly addressable flash storage that holds the operating system and embedded software, that is, the program

5

10

15

20

25

25

code comprising the instructions that the processor must execute to operate. The flash memory, which may be conventional flash memory that is available from a variety of vendors, need not be of the removable type, as the Program Code Flash Memory 107 is not intended to be removed from the system by the camera user.

5 The Processor 106 itself, in the presently-preferred embodiment, comprises a 32-bit RISC ARM Processor designed by ARM Limited of Maidenhead, UK. ARM licenses its designs to semiconductor partners for manufacture, supply, and support; for a list of ARM licensees, see e.g., <http://www.arm.com/Partners/>. The ARM processor has an efficient instruction set that is ideal for performing cyclical functions quite rapidly and includes sufficient bandwidth for transferring large amounts of data quickly (e.g., for performing Huffman coding on a large amount of data). Additionally, the processor is a dedicated processor, without the overhead of a substantial number of peripherals. These features make 10 the processor attractive for use in a digital camera embodiment.

15 For a camera embodiment, the device will, in general, be expected to include an interface that is capable of receiving input from users. Keypad and Controls 108 are conventional inputs that support user input. Similarly, the Direct View Display (“Viewfinder”) 109 is a direct view LCD (liquid crystal display) that provides feedback to the user or camera operator. During photography mode, the Viewfinder 109 replaces the plastic viewfinders and LCD panels found on most digital cameras and provides the most accurate 20 real-time representation of the scene visualized by the sensor. The Viewfinder 109 overlays simple icons onto the image to indicate the status of various camera settings. The Viewfinder 109 fits inside an eyepiece which keeps sunlight out and allows the operator to visualize the scene in any lighting conditions. During preview mode, the Viewfinder 109 shows previews of the captured photos and allows the operator to delete unwanted photos or tag photos for 25 wireless transmission. Thus for a camera embodiment, the Viewfinder 109 is used to provide a representation of the image that is being captured, in preview and/or post-capture fashion.

In order to provide the display image to the Viewfinder 109, the Sensor 101 is subsampled at a rate to create a version of the image appropriate for display. During preview processing, the system continuously captures the sensor mosaic and sub-samples the resulting

mosaic for preview purposes. A histogram of the sampled luminosity is fed into a “linearization” filter to produce a balanced dynamic range for best optical perception. The scaled and “linearized” image is then displayed on the viewfinder module. The histogram data is then adjusted to match the preview image for use in linearizing the next image. The cycle is repeated continuously to provide a real-time viewfinder mechanism. The Viewfinder 109 itself typically operates in conjunction with a display controller and a frame buffer (not shown), both of which may be integrated within the display component itself.

Both the Keypad and Controls and Display components, which may be conventional in nature, interface directly with the Processor 106 through general I/O (e.g., I/O Bus). Typically, such devices communicate with the microprocessor through means of interrupt requests (IRQ). Both the Keypad and Controls and Display components are available from a variety of vendors. Examples include Sharp, Toshiba, and Citizen of Japan, Samsung of South Korea, and Hewlett-Packard of Palo Alto, CA. More customized displays are available from Displaytech, Inc. of Longmont, CO. For an embodiment that does not need to interact with users, such as a surveillance camera, the foregoing components may be eliminated.

Additionally for a camera embodiment, it is desirable for the device to include an interface for standard peripheral devices, such as a detachable flash device. This may be provided by Hot Shoe (Accessory) Interface 110, which is a general I/O port that may comprise a serial interface of a conventional design that the camera uses to interface to its accessories via the Hot Shoe Interface. In this manner, a flash accessory can be clipped onto the camera via the Hot Shoe Interface for added illumination.

The Hot Shoe Interface 110 combines a Serial Peripheral Interface (SPI) with a multiplexed I/O bus which provides a plug-and-play interface to a family of accessories. These accessories may include, in addition to a flash unit, a wireless holster for cellular (e.g., Motorola) phones, extra film backs for compatibility with format digital film (e.g., Sony Memory Stick or SmartMedia), a USB cradle, an RJ-11 modem cradle, a wireless cellular module, extender cables, and the like. In the currently-preferred embodiment, the interface is based on the I²C-standard serial interface, which supports logic allowing the device to sense

I²C-compatible devices that are attached to the port. I²C, which stands for Inter IC Communication, is a serial bi-directional communication protocol created by Philips Semiconductor (subsidiary of Philips Electronics, based in The Netherlands) and is used for communication between integrated circuits. Most systems have one master and several slaves that communicate using only two wires. Every device has its own identification code. If that code is sent by the master only that device will respond with an acknowledgement. After the acknowledgement, the data to be communicated is sent or received by the master. Further information about the I²C communication protocol is available from Philips Electronics of The Netherlands. As with the Keypad and Controls 108 and Direct View Display or Viewfinder 109, the Hot Shoe Interface 110 itself is not required for implementing the image capturing and processing methodology of the present invention. In the specific embodiment of a consumer product such as a camera, though, these components typically would be included.

The system includes Digital Film Flash Memory 111, which serves as the "digital film" for the system for storing compressed images. The Flash Memory 111 may comprise available flash memory removable media, such as CompactFlash, DataFlash, and Sony Memory Stick, typically in a 16 MB or larger size. Available vendors for flash memory include, for example, SanDisk of Sunnyvale, CA or Sony of Japan. Alternatively, the Flash Memory 111 may be affixed directly (i.e., non-removable) to the system 100. In such an embodiment, the additional bulk associated with a removable media cartridge holder and its accompanying interface may be avoided. Those skilled in the art will appreciate that the system 100 may incorporate other non-volatile memory configurations and designs that readily accommodate the image capture and processing methodology of the present invention. In general, for a consumer device embodiment, one should choose media that accommodates on the order of 100 compressed images or more.

The camera embodiment is powered by a single CR-123 lithium battery (not shown), provided with instant-on capability. Due in part to the distributed image processing approach of the present invention (presented below), the camera has significant power

savings over other camera designs. This gives the device not only a size and weight advantage over other cameras but also a battery life advantage.

For connectivity, the system includes a wireless holster, a USB cradle, and a modem cradle. The wireless holster physically connects the camera to a cellular phone (e.g.,
5 Motorola cellular phone) and interfaces the Hot Shoe Interface to the phone's external accessory plug. The camera can be easily pulled out of the holster for use and clipped back in for transmission. Detection of the holster and phone signal is automatic to allow for hands-free transmission and there is no risk of corruption due to interruption by either loss of signal or unclipping. The camera clips into the USB cradle through the Accessory Hot-Shoe
10 Interface 110 to provide rapid photo interchange to a personal computer equipped with a standard USB port. The USB cradle acts a USB slave device and therefore requires no batteries or power supply for operation and instead draws its power from the PC. The camera can also clip into a modem cradle through the Hot Shoe Interface. The modem cradle allows the camera to transmit images to the PhotoServer via a land line connection (e.g., 33.6KBps)
15 via a standard RJ-11 phone jack. The modem cradle is powered by the battery in the camera.

The specifications for the currently-preferred camera embodiment may be summarized as follows.

TABLE 1: Miniature Wireless Digital Camera Specifications:

Sensor:	1.3 Mega-Pixel Color CMOS
Optics:	18mm Fixed Focal Length, Fixed Aperture
Exposure Control:	Automatic, Macro Mode, Indoor/Outdoor Mode
Processor:	ARM 32-bit RISC
25 Chipset:	Image Processor (Lightsurf PhotonOne)
Memory:	4Mbytes DRAM + 128Kbytes SRAM
Digital Film:	16Mbytes Internal Flash Film
File Format:	Progressive Photograph Format (PPF)

Wireless Protocol: Communication protocol, such as packet-based TCP/IP, WAP, or the like

Battery: CR-123

Accessory Interface: Accessory Hot-Shoe

5 Accessories: Flash Unit, Extra Film Back, Motorola Cellular Holster, USB Cradle, Modem Cradle

B. Basic computer hardware (e.g., for desktop and server computers)

Portions of the present invention may be implemented on a conventional or general-purpose computer system, such as an IBM-compatible personal computer (PC) or server computer. Fig. 1B is a very general block diagram of an IBM-compatible system 100, which is adapted to include portions of the distributed image processing of the present invention. As shown, system 150 comprises a central processor unit(s) (CPU) 151 coupled to a random-access memory (RAM) 152, a read-only memory (ROM) 153, a keyboard 156, a pointing device 158, a display or video adapter 154 connected to a display device 155, a removable (mass) storage device 165 (e.g., floppy disk), a fixed (mass) storage device 166 (e.g., hard disk), a communication port(s) or interface(s) 160, a modem 162, and a network interface card (NIC) or controller 161 (e.g., Ethernet). Although not shown separately, a real-time system clock is included with the system 150, in a conventional manner.

CPU 151 comprises a processor of the Intel Pentium® family of microprocessors. However, any other suitable microprocessor or microcomputer may be utilized for implementing the present invention. The CPU 151 communicates with other components of the system via a bi-directional system bus (including any necessary I/O controller circuitry and other “glue” logic). The bus, which includes address lines for addressing system memory, provides data transfer between and among the various components. Description of Pentium-class microprocessors and their instruction set, bus architecture, and control lines is available from Intel Corporation of Santa Clara, CA. Random-access memory 152 serves as the working memory for the CPU 151. In a typical configuration, RAM of sixteen megabytes or more is employed. More or less memory may

be used without departing from the scope of the present invention. The read-only memory (ROM) 153 contains the basic input/output system code (BIOS) -- a set of low-level routines in the ROM that application programs and the operating systems can use to interact with the hardware, including reading characters from the keyboard, outputting characters to printers, and so forth.

Mass storage devices 165, 166 provide persistent storage on fixed and removable media, such as magnetic, optical or magnetic-optical storage systems, or flash memory, or any other available mass storage technology. The mass storage may be shared on a network, or it may be a dedicated mass storage. As shown in Fig. 1B, fixed storage 166 stores a body of program and data for directing operation of the computer system, including an operating system, user application programs, driver and other support files, as well as other data files of all sorts. Typically, the fixed storage 166 serves as the main hard disk for the system and stores application software implementing the PhotoServer (PhotoDesktop) component described below.

In basic operation, program logic (including that which implements methodology of the present invention described below) is loaded from the storage device or mass (fixed) storage 166 into the main (RAM) memory 152, for execution by the CPU 151. During operation of the program logic, the system 150 accepts user input from a keyboard 156 and pointing device 158, as well as speech-based input from a voice recognition system (not shown). The keyboard 156 permits selection of application programs, entry of keyboard-based input or data, and selection and manipulation of individual data objects displayed on the display screen 155. Likewise, the pointing device 158, such as a mouse, track ball, pen device, or the like, permits selection and manipulation of objects on the display screen. In this manner, these input devices support manual user input for any process running on the system.

The computer system displays text and/or graphic images and other data on the display device 155. Display device 155 is driven by the video adapter 154, which is interposed between the display device 155 and the system. The video adapter 154, which includes video memory accessible to the CPU, provides circuitry that converts pixel data

5 stored in the video memory to a raster signal suitable for use by a cathode ray tube (CRT) raster or liquid crystal display (LCD) monitor. A hard copy of the displayed information, or other information within the system 150, may be obtained from the printer 157, or other output device. The printer 157 may include, for instance, an HP Laserjet® printer (available from Hewlett-Packard of Palo Alto, CA), for creating hard copy images of output of the

system.

10 The system itself communicates with other devices (e.g., other computers) via the network interface card (NIC) 161 connected to a network (e.g., Ethernet network), and/or a modem 162 (e.g., 56K baud, ISDN, DSL, or cable modem), examples of which are available from 3Com of Santa Clara, CA. The system 150 may also communicate with local occasionally-connected devices (e.g., serial cable-linked devices) via the communication (“comm”) interface 160, which may include an RS-232 serial port, a Universal Serial Bus (USB) interface, or the like. Devices that will be commonly connected locally to the comm interface 160 include laptop computers, handheld organizers, digital cameras, and the like.

15 IBM-compatible personal computers and server computers are available from a variety of vendors. Representative vendors include Dell Computers of Round Rock, TX, Compaq Computers of Houston, TX, and IBM of Armonk, NY. Other suitable computers include Apple-compatible computers (e.g., Macintosh), which are available from Apple Computer of Cupertino, CA, and Sun Solaris workstations, which are available from Sun Microsystems of Mountain View, CA.

20 The above-described system 150 is presented for purposes of illustrating the basic hardware underlying desktop and server computer components that may be employed in the system of the present invention. For purposes of discussion, the following description will present examples in which it will be assumed that there exists a “server” or remote device having information of interest to the ultimate end-user. The present invention, however, is not limited to any particular environment or device configuration. In particular, a server distinction is not necessary to the invention, but is used to provide a framework for discussion. Instead, the present invention may be implemented in any type of computer

system or processing environment capable of supporting the methodologies of the present invention presented in detail below.

C. Basic system software

Illustrated in Fig. 2, a computer software system 200 is provided for directing the operation of the computer system 150. Software system 200, which is stored in system memory 152 and on fixed storage (e.g., hard disk) 166, includes a kernel or operating system (OS) 210. The OS 210 manages low-level aspects of computer operation, including managing execution of processes, memory allocation, file input and output (I/O), and device I/O. One or more application programs, such as client application software or “programs” 201 (e.g., 201a, 201b, 201c), including image processing software, may be “loaded” (i.e., transferred from fixed storage 166 into memory 152) for execution by the system 150.

System 200 includes a graphical user interface (GUI) 215, for receiving user commands and data in a graphical (e.g., “point-and-click”) fashion. These inputs, in turn, may be acted upon by the system 150 in accordance with instructions from operating system 210, and/or client application module(s) 201. The GUI 215 also serves to display the results of operation from the OS 210 and application(s) 201, whereupon the user may supply additional inputs or terminate the session. Typically, the OS 210 operates in conjunction with device drivers 220 (e.g., “Winsock” driver) and the system BIOS microcode 230 (i.e., ROM-based microcode), particularly when interfacing with peripheral devices. OS 210 can be provided by a conventional operating system, such as Microsoft® Windows 9x, by Microsoft® Windows NT, or by Microsoft® Windows 2000, all available from Microsoft Corporation of Redmond, WA. Alternatively, OS 210 can also be an alternative operating system, such as IBM OS/2 (available from IBM of Armonk, NY) or Macintosh OS (available from Apple Computer of Cupertino, CA).

Distributed Digital Image Processing

A. Basic design consideration

The basic approach adopted by the present invention is to adopt techniques for reducing the amount of processing power required by a given digital camera device and for 5 reducing the bandwidth required for transmitting image information to a target platform.

Every digital imaging device is inherently connected. Consider, for instance, a digital camera. It does not necessarily appear to be a "connected" device, as it may initially give the appearance of an end-to-end solution, where one is capturing an image, processing that image, and then storing it on digital film. In reality, however, the true end product is some 10 type of output, such as a printed image used in a document. As it turns out, somewhere along the way the image is taken off the device and transmitted to yet another computing device, such as a server or desktop computer, where, for instance, the image may be cropped, touched up, or otherwise processed. Therefore, a digital image -- due to the fact that it is digital, is inherently related to all other types of computing devices that can handle images. Given that 15 environment in which digital cameras exist, there is an opportunity to take advantage of other processing power that is eventually going to come into contact with the images that are produced by the digital imaging device ("imager"). More particularly, there is an opportunity to defer and/or distribute the processing between the digital imager itself and the target platform that the digital imager will ultimately be connected to, either directly or indirectly. 20 Therefore, rather than attempting to invent a revolutionary way to get better hardware performance (i.e., better silicon technology) or a revolutionary compression technique, the approach of the present invention is to decrease the actual computation that occurs at the digital imager: perform a partial computation at the digital imager device and complete the computation somewhere else -- somewhere where time and size are not an issue (relative to 25 the imager). In other words, recognizing that the images captured by a digital camera will typically end up on another computing device, the approach of the present invention is to take advantage of that fact by "re-architecting" the digital camera to defer resource-intensive computations, thereby substantially eliminating the processor requirements and concomitant battery requirements for digital cameras. Further, the present invention adopts an image

strategy which facilitates transmission of images, thereby facilitating the wireless transmission of digital camera images.

For purposes of determining how to defer and/or distribute processing, the overall process of digital imaging, from capturing a light image to storing a digital representation of that image, may itself be viewed as comprising various subprocesses. Once individual subprocesses are identified, one can investigate various approaches for deferring and/or distributing those subprocesses to other devices. Consider, for instance, a decision to defer image compression. Such an approach entails immediate problems, however. The digital camera must have sufficient resources to store, at least temporarily, uncompressed images, which tend to be quite large in size. Although storage technology (e.g., flash memory) can be expected to have ever-increasing capacity, present-day storage technology makes that approach unattractive, if not impractical. Another difficulty posed by that approach is that the digital camera must exchange images in an uncompressed format with the target device (e.g., desktop computer). Again, such an approach is unattractive since it would require the user to spend an inordinate amount of time transferring images to the target device, given the limited bandwidth that is commercially feasible for downloading pictures from a digital camera. Therefore, an approach of eliminating compression is not attractive, unless one can somehow obtain massive storage and bandwidth capacity for the digital camera. The immediate challenge posed by a deferred/distributed processing approach, therefore, is how one can accomplish the approach in a manner that does not contradict the ultimate goal of obtaining quality digital images on a portable digital camera. To address this challenge, the entire imaging process, including its individual elements, is examined in order to figure out how the process can be reproduced in a manner that compute-intensive portions of the process are performed somewhere other than the digital camera itself, but done so in a manner that does not compromise the image information necessary for reconstructing a digital image of acceptable quality.

5

10

15

20

25

B. Distributed image processing

As illustrated in Fig. 1B, the imaging process approach of the present invention includes a rapid foreground process to capture and compress the image (e.g., one-second cycle) and a slower background process to further compress and transmit the image. The foreground process is optimized for speed to facilitate continuous rapid snapshots while the background process is optimized for power. As shown, the two-stage processing mechanism is assisted by an imaging processing server, the "PhotoServer," which typically includes Internet connectivity.

The first stage, the foreground stage, is performed in the camera itself. This stage produces a highly-compressed image based on wavelet transform technology. This image is stored on the 16MB of digital film inside the camera. The image is then transmitted to the PhotoServer (target platform) via a packet-based protocol as a Progressive Photograph Format (PPF) file. Suitable protocols include, for instance, Transmission Control Protocol/Internet Protocol (TCP/IP) and Wireless Application Protocol (WAP). For a description of TCP/IP, see e.g., Parker, T. et al., *TCP/IP Unleashed*, Sams Publishing, p. 33 et. seq., the disclosure of the reference in its entirety being hereby incorporated by reference. For a description of WAP, see e.g., Mann, S., *The Wireless Application Protocol*, Dr. Dobb's Journal, pp. 56-66, October 1999, the disclosure of which is hereby incorporated by reference.

The PPF mechanism allows selective transmission of varying degrees of photographic significance while maintaining the remaining elements of significance on the digital film in the camera. A lower quality image transmitted to the server can later be upgraded to a higher quality version simply by synchronizing the remaining elements of significance stored in the PPF file. The second stage of the process, the background stage, is performed on the PhotoServer. This stage completes the image processing cycle and produces a high quality, color-balanced, compressed image in a standard file format such as JPEG. A similar second stage process also exists in a desktop implementation, the PhotoDesktop, for customers who wish to transfer images to a desktop computer (as the

target platform) using a USB (universal serial bus) cradle accessory or other communication link.

Detailed construction and operation of the foregoing is perhaps best described by way of comparison with conventional digital image processing. Fig. 3A illustrates on its left-hand side a high-level process or methodology 310 that comprises the individual processes (i.e., subprocesses) or steps routinely employed for digital image processing. The digital imaging process or sequence 310 begins with image capture, as represented by capture block or capture process 311. This is the process of capturing light data (image) from a sensor and, in response, generating digital representations of that data (i.e., digital light levels based on the captured image). This is, in effect, an in-memory image of the light which has entered the camera's lens and struck the camera's CCD or CMOS sensor. It is interesting to note that at this point the digital camera has only captured light levels. Color information per se does not yet exist. Instead, color information, which must be inferred, is not determined yet at this point in the digital imaging process.

The capture process 311 is conventionally followed by a color interpolation (transformation) process 313, where color information may indeed be determined and applied to the image. In practice, the camera is able to infer color information based on captured luminance data and information that it knows about the individual pixels and where they lie within a matrix of color (pixels) that cover the surface of the camera's sensor. This information is now applied to the image by the color interpolation process 313, which is a compute-intensive process.

Because the human eye is more perceptive to certain colors than others, further processing of the image is required. The standard color space that the device "sees" the image in (e.g., RGB color space or model) is not necessarily the way that the human eye would view the image. For instance, the human eye has a certain distribution of retinal cones that are more sensitive to certain wavelengths of light. Therefore, in an effort to better match that expected by the human eye, YUV transformation process 315 maps or translates the image (e.g., RGB-based image) into YUV color space, a color model which takes into account luminance and chrominance. In YUV, Y is the luminance component, and U and V

are the color or chrominance components. Luminance serves as a quantitative measure of brightness. Chrominance, on the other hand, reflects the difference in color quality between a color and a reference color that has an equal brightness and a specified chromaticity. In essence, the YUV transformation process 315 is a matrix transformation. Here, the red, green, and blue (RGB) values that apply to a particular piece of pixel data are multiplied by a vector which, in turn, translates the values into YUV color space. Although the individual transformation itself is not particularly complicated, the YUV transformation process 315 is applied to every pixel of the image and, thus, consumes a lot of processing cycles. Accordingly, the YUV transformation process 315 itself is also compute-intensive.

Now, the image may be compressed as represented by compression process 317. The basic approach applied in image compression is to prioritize the image data according to how a human eye would normally see it. In lossy image compression technique (e.g., JPEG), the chrominance levels that are less important to the human eye are compressed out. Luminance, which the human eye is more sensitive to, is given priority in the compression. Lossy techniques, which function largely by eliminating information that is the least significant to the human eye, are described in the technical, trade, and patent literature. See e.g., Nelson, M. et al., *The Data Compression Book*, Second Edition, Chapter 11: Lossy Graphics Compression (particularly at pp. 326-330), M&T Books, 1996. Also see e.g., *JPEG-like Image Compression* (Parts 1 and 2), Dr. Dobb's Journal, July 1995 and August 1995 respectively (available on CD ROM as *Dr. Dobb's/CD Release 6* from Dr. Dobb's Journal of San Mateo, CA). The disclosures of the foregoing are hereby incorporated by reference.

After the foregoing lossy image capture process, the now-compressed image may be stored on persistent media. As indicated by processing block 318, conventional flash memory (or other suitable media) may be employed for this purpose, such as any one of the flash memory varieties previously mentioned. The end result is a compressed JPEG file 319, as shown.

The design goal of digital cameras today is to generate at the digital camera itself complete color images that are compressed in a standard image format (e.g., JPEG).

This sequence, however, incurs multiple compute-intensive processes, including the color transformation process 313, the YUV interpolation process 315, and the compression process 317, with the end result being a relatively large image that is not well-suited for wireless transmission. In accordance with the teachings of the present invention, however, the foregoing design goal is not adopted. Note that the very color images that have been processed, at great computational expense, into a standard image format at the digital camera will ultimately be transferred to another device -- another piece of computing hardware. If one assumes, for instance, that the images do not necessarily have to be exactly a color JPEG (or other standard image format) while stored on the digital camera, but that the images will be a color JPEG ultimately (e.g., target device), then one can begin to defer some of the processes required for processing digital images. In accordance with the present invention, image processing of a digital camera is performed in such a manner so as to retain the advantage of compression at the digital camera but remove the compute-intensive components of the process, so that they may be performed elsewhere (other than the digital camera itself). This process will now be described in further detail.

The approach of the present invention exploits the fact that there is really no such thing as "color sensitivity" on the digital camera. Instead, the camera uses a color inference transformation that employs a mosaic pattern of color pixels overlaid on a light-level sensor (e.g., CCD or CMOS sensor). For instance, as light shines through a red tile on the mosaic, it lights up the light-sensing element behind it. That element, in turn, records the luminosity observed. The digital camera may then employ its internal knowledge of the mosaic pattern -- the specific arrangement of red, green, and blue pixels (of which there are a variety of standard patterns) -- to infer the actual color. Each pattern itself tends to be a small, repeating pattern that is capable of being represented with a small amount of data. For example, a Bayer pattern will consist of a 2x2 pixel section (four pixels total) of two green pixels on one diagonal and a red and blue on another diagonal that is repeated over and over again. A simplistic approach to using the light information arriving at a particular pixel section is to merge together (i.e., matrix transformation) the information to produce four pixels having the same RGB level, at one quarter the resolution, but with accurate color data.

Another approach is to take into account the luminosity observed at each of the pixels, so that not only is color information incorporated into the image processing but also the sharpness that is being perceived by each sensor as well.

The color interpolation process does not itself enhance the image data.

5 Although it is certainly required for ultimately rendering a color image, it itself need not necessarily be performed at the digital camera and can therefore be deferred. Stated differently, if the knowledge of the color pattern can be placed somewhere else and the color image need not be completely generated right away (i.e., at the digital camera), then the step or process of transforming the originally-captured image into a color image may be deferred.

10 In accordance with the present invention, the color interpolation and transformation process is in fact deferred altogether. Instead, the R, G, B color planes are separated and compressed. The compressed images are packaged into a single stream with header information to identify the individual bit-streams. The combined bit-stream may then be transmitted to the target device, with a small descriptor of what Bayer pattern should be applied either being transmitted to or being assumed by the target device.

15 The right-hand side of Fig. 3A illustrates a preferred methodology 320 for digital image processing in accordance with the present invention. At the outset, an image is captured by capture process 321, in a manner similar to that previously described for capture process 311. At the moment the shutter button is depressed, the sensor captures a full detail mosaic in two phases. The first phase is captured with the mechanical shutter closed, the second with the shutter open. Both phases happen in rapid succession, with the first being used to normalize the black level of the second. The mosaic is then fed into a linearization filter using the coefficients from the last preview frame prior to shutter click and serialized to DRAM. The image is also scaled to match the operator-selected image capture resolution.

20 Any aberrant pixel data should fall outside of the dynamic range provided by the histogram and consequently be canceled out.

25 Next, the color interpolation or transformation process is entirely avoided. Instead, the methodology 320 immediately moves to extraction of the color planes, shown as color plane separation process 323, followed by a wavelet transform process 330 to prioritize

information in the color planes. Here, the separated color planes are fed into a wavelet transform image -- that is, a preferably hardware-implemented (for performance) wavelet transform process. Over a series of repeated cycles, the wavelet engine transforms the luminosity image in place in DRAM.

5 The wavelet transform process itself may be thought of as a process that applies a transform as a sequence of high- and low-pass filters. In operation, the transformation is applied by stepping through the individual pixels and applying the transform. This process, which creates an image that contains four quadrants, may for instance be performed as follows. First, a high-pass transform then a low-pass transform is 10 performed in the horizontal direction. This is followed by a high-pass transform then a low-pass transform performed in the vertical direction. The upper-left quadrant is derived from a low-pass horizontal/low-pass vertical image; the lower-left quadrant comprises a high-pass horizontal/low-pass vertical image; the upper-right quadrant comprises a low-pass horizontal/high-pass vertical image; and the lower-right quadrant comprises a high-pass horizontal/high-pass vertical image. The result of this is that the information most important 15 to the human eye (i.e., the information that, from a luminosity or black/white perspective, the human eye is most sensitive to) is in the high-priority “low/low” quadrant, that is, the upper-left quadrant which contains the low-pass horizontal/low-pass vertical image. Most of the information in the other three quadrants, particularly the lower-right quadrant, is 20 fundamentally zero (when based as an onset of a center frequency), that is, image information that is least perceived by the human eye. Thus, the low/low quadrant is considered the highest-priority quadrant, with the remaining quadrants being considered to be of much lower priority.

25 The transform is a completely reversible process, such that the original image (luminosity record) may be restored without loss of information. In practice, however, some information is lost as a result of the process being performed digitally, as digital computing devices are of course only able to perform real number math with finite, not infinite, precision. Nevertheless given enough digital significance (that is typically available with processors today), this loss is imperceptible to the human eye. The human eye is perceptive

only to a bit depth in the range of about five to six significant bits of image data (based on a certain center frequency). As a result, processing the image with even modest processing capability (e.g., a bit depth of 16 significant bits) generates a transform that is reversible in a manner that is not perceptible to the human eye. Here, the image data is arranged without any reliance on color in such a way that the information most important to the human eye is one-quarter of its original size. If desired, the best-perceived quadrant (e.g., the upper-left quadrant for the example sequence above) may be used as a basis to provide the user with a black and white image at the digital camera (e.g., for preview purposes).

In basic operation, the transform process consists of processing the image as a whole in a stepwise, linear fashion. For instance, when processing the image in a horizontal direction, one would take a horizontal vector of image data (e.g., seven horizontal neighboring pixels) and multiply that by a predetermined set of coefficients (e.g., seven coefficients for a seven-pixel vector). This yields a single pixel value. Then the process continues in a sliding-window fashion by shifting over by some number of pixel(s) (e.g., two pixels), for processing the next vector of seven horizontal neighboring pixels. Further description of the wavelet transform process may be found, for instance, in the technical and trade literature. See e.g., Pigeon, S., *Image Compression with Wavelets*, Dr. Dobb's Journal, August 1999, pp. 111-115. The disclosure of the foregoing is hereby incorporated by reference, for all purposes.

As illustrated in Fig. 3B, the wavelet transform process may be repeated multiple times, if desired. When repeated, the process of applying high- and low-pass filters is repeated for the low/low quadrant of the then-current image (i.e., the prior result of high-pass horizontal and vertical filtering), again generating a four-quadrant image. For instance, as shown in Fig. 3B, the wavelet-transformed image 370, which has already undergone a first pass of the wavelet transform, is subjected to another pass of the wavelet transform process to generate wavelet-transformed image 380 -- that is, an image that has undergone two rounds of wavelet transformation. The process may continue in this fashion, for example, generating wavelet-transformed image 390. Each time, the subsequent pass is performed on the prior-resulting low/low quadrant. Those skilled in the art will recognize

that other quadrants could also be decomposed in a similar manner. This process may continue recursively until the desired transformed image is obtained. Whether the image is transformed with a single pass or multiple passes, the end result is still a wavelet-transformed image. This image is “quantized” (i.e., reduced in bit-depth) by dividing the wavelet 5 coefficients (i.e., the numerical value of the pixels in the wavelet-transformed image) by a quantization scale factor. The quantization can differ from one band to another. The quantizations step sizes will be included in compressed bit-stream and will be used by the decompression system (e.g., on the desktop/server) to reverse the above process. Note that quantization and dequantization leads to loss of precision in the wavelet data and represents 10 the lossy part of the compression. After quantization, the wavelet coefficients are compressed losslessly by one of several generic binary compression techniques (e.g., bit-plane decomposition of bands, followed by arithmetic coding).

After generating the wavelet-transformed image, therefore, the preferred methodology 320 of the present invention proceeds to apply compression to the image. At this point, the image information (i.e., all quadrants and subquadrants) can be compressed as if it were fundamentally just a normal binary file. Thus, one can apply a simple, conventional compression, as a compute-efficient compression process, as indicated by compression process 340. In a preferred embodiment, the compression is performed in succession stages. First, run-length encoding (RLE) is applied to compress the image data. RLE itself is a 15 simple, well-known technique used to compress runs of identical symbols in a data stream. The insignificant regions of the image data (i.e., the low-priority quadrants) tend to be predominantly centered around a single value; these can be compressed substantially. When applying run-length encoding to this type of information, for instance, one gets extremely 20 long runs of similar data. The image is serialized to flash memory during the encoding process to free the DRAM for the next image capture. The entire cycle from image capture through stage one compression and serialization to flash is rapid (e.g., less than one second) for the highest quality mode. The camera is then ready to take another photograph. RLE, 25 which typically encodes a run of symbols as a symbol and a count, is described in the patent, technical, and trade press; see, e.g., Zigon, Robert, *Run-Length Encoding*, Dr. Dobb’s

Journal, February 1989 (available on CD ROM as *Dr. Dobb's/CD Release 6* from Dr. Dobb's Journal of San Mateo, CA), the disclosure of which is hereby incorporated by reference. In addition to RLE, the methodology 320 may include discarding low priority data in order to provide more aggressive lossy compression.

5 This target result may then, in turn, be further compressed using Huffman coding, for generating a final compressed luminosity record 350 that is suitable for storage on a digital camera and for wireless transmission. Huffman coding is a method of encoding symbols that varies the length of the symbol in proportion to its information content. Symbols with a low probability of appearance are encoded with a code using many bits, 10 while symbols with a high probability of appearance are represented with a code using fewer bits. Huffman coding is described in the patent, technical, and trade press; see, e.g., Nelson, M. et al., *The Data Compression Book*, Second Edition, Chapters 4 and 5, M&T Books, 1996, the disclosure of which is hereby incorporated by reference.

15 The wavelet transform-based compression used in the digital camera 100 achieves significantly better image quality than traditional JPEG compression used in other digital cameras. The image comparisons presented in Figs. 3C - I illustrate this. The standard reference image for compression algorithms is the Lena image, shown in Fig. 3C in original uncompressed detail. The following image sets illustrate the resulting relative image degradation of wavelet and JPEG techniques for varying compression ratios. At an 20 ultra-aggressive 48:1 compression ratio, the Lena image is still of reasonable quality using wavelet transform (Fig. 3E) while JPEG (Fig. 3D) has generated unacceptable pixelization. At a compression ratio of 32:1, the Lena image is showing better edge detail with wavelet transform (Fig. 3G) while JPEG (Fig. 3F) is still largely pixelized and unacceptable. At a compression ratio of 16:1, the fast quality compression ratio, wavelet transform (Fig. 3I) has 25 produced a good quality image with good edge definition and few noticeable artifacts. JPEG (Fig. 3H), on the other hand, is barely approaching an acceptable image comparable with a wavelet ratio of 32:1 or more. Thus, the foregoing demonstrates that the wavelet transform-based compression technique produces far better shade continuity and edge detail than the equivalent JPEG.

Thus as described above, the camera-implemented portion of image processing (i.e., methodology 320) foregoes color processing. Instead of performing YUV transformation, the methodology performs wavelet transform compression on an image comprising a luminosity record. Further, JPEG-style compression, which is fairly compute-intensive, has been removed. Instead, the methodology 320 applies generic binary compression (e.g., run-length encoding and Huffman coding), which is far less compute-intensive. Note in particular that, up to this point, image compression in accordance with the present invention has been performed in a manner which is largely lossless, not lossy. Loss of image information at this point, which is quite small, is due only to digital rounding errors. If desired, however, additional compression techniques, including lossy ones, may be applied (e.g., at additional compression process 340). For instance, the image may be further compressed by reducing the bit depth in the low priority quadrants.

The end result is that the amount of processing necessary to go from a captured image to a compressed record of the captured image (i.e., a record suitable for storage on the digital camera) is substantially less than that necessary for transforming the captured image into color and then compressing it into a color-rendered compressed image. Further, the resulting compressed luminosity record, because of its increased compression ratios (e.g., relative to conventional JPEG), facilitates wireless (or other limited bandwidth) transfer of images to target platforms.

The compressed luminosity record 350 is of course optimized for generation and storage on a digital camera, not for viewing by the user. Thus at the point where the compressed luminosity record 350 is transferred to another computing device (e.g., images downloaded to a desktop computer), image processing crosses over the distributed boundary to continue image processing on the target platform. In the currently-preferred embodiment, this is done via wireless transmission. Whenever the camera is connected to the cellular holster via the Hot-Shoe clip, a background process is initiated to transmit any pending compressed PPF photographs to the PhotoServer for final processing. The process is transparent to the user, requiring no operator intervention and can occur while the camera is

in low power mode. Using WPTP as the transport layer, the process can be interrupted at any time without worry of any data corruption or need to re-send already transmitted packets.

Estimates for wireless transmission times follow in the tables below. These estimates are shown with varying operator-selectable image resolution and varying operator selectable image quality. Image quality is a factor of compression ratio. As compression ratios increase, more loss of significant image data occurs.

TABLE 2: Wireless transmission times assuming a 10 Kbps transmission rate

Resolution	Photographic	Standard	Internet
Quality	1280 x 1024	1024 x 768	512 x 384
High (4:1)	255 seconds	153 seconds	38 seconds
Standard (8:1)	126 seconds	76 seconds	19 seconds
Fast (16:1)	63 seconds	38 seconds	8 seconds

TABLE 3: Wireless transmission times assuming a 56 Kbps transmission rate

Resolution	Photographic	Standard	Internet
Quality	1280 x 1024	1024 x 768	512 x 384
High (4:1)	51 seconds	31 seconds	8 seconds
Standard (8:1)	23 seconds	15 seconds	4 seconds
Fast (16:1)	13 seconds	8 seconds	2 seconds

As an option, an operator can also transmit the PPF photographs to a personal computer via the USB cradle. This process employs the same packet-based communication protocols except that it happens over a wire-line connection and in the foreground.

Photographs transmitted to the PhotoServer or to the PhotoDesktop can be synchronized using sections of the PPF file. Synchronization is the act of supplying any supplemental data to images in order to enhance them to the maximum quality PPF record available. For instance, synchronization of a fast quality PPF file and a high quality PPF file of the same

image will result in enhancement of the fast quality image to high quality.

The Progressive Photograph Format (PPF) itself comprises a sequence of sequential image sections ordered by decreasing optical significance. The first section is the most significant image data and represents a complete fast quality version of the image. This is followed by sections two and three which contain subsequent detail data to enhance the image to normal quality and high quality respectively. Using the PPF approach, a fast quality image can be transmitted to the PhotoServer, taking maximum advantage of transmission data size and speed. The server image can then be synchronized with the remaining components of the PPF file at a later time to restore the image to its original maximum quality. With this unique approach, the operator does not have to sacrifice image quality in order to maximize wireless throughput.

Now, the remainder of image processing can be performed at the target platform (e.g., server or desktop computer) in a straightforward manner, without the size and portability constraints that are imposed on the digital camera. Moreover, one can apply all of the processor capability of the target platform. Note, however, that the foregoing approach may be modified so that the image is (optionally) color processed at the digital camera (e.g., for viewing as a color JPEG file), yet transmitted as a PPF file, thus preserving the high-compression benefit for wireless transmission.

Fig. 4A provides an overview of the completion of image processing at the target platform. The PhotoServer receives highly compressed PPF files from the digital camera and completes the image processing cycle. In the decompression phase, a decompressed image is reconstructed from the PPF file. The resulting image is then run through an artifact reduction filter which compensates for artifacts introduced by the camera during the compression process. The result is then arranged into the original sensor color mosaic. The image is then processed by the color interpolation engine, which removes the mosaic pattern and produces a high quality color image. The resulting image is in the RGB color space. Next the color characterization profile of the specific camera (recorded at factory assembly time) is used to balance the colors in the image to match human color perception. This stage makes up for the differences in how the camera sensor and the human

eye see color. The image enhancement phase is optional. In this phase the balanced color image is processed to reduce red-eye artifacts, to enhance contrast, to harden and smooth edges, or even to interpolate to a higher resolution. Finally, the image is again compressed. The resulting output is an industry standard, high quality color image file such as JPEG, JPEG 2000 or FlashPix.

Fig. 4B illustrates specific method steps involved at the target platform. First, the decompression process 410 decompresses the compressed luminosity record (e.g., reversing the Huffman coding and RLE compression). Thereafter, the wavelet transform is reversed, for restoring the uncompressed luminosity record. This is illustrated by the inverse wavelet transform process 420. Reversing the wavelet transform process yields an uncompressed luminosity record 430 -- that is, a record of that which was originally sent from the camera's sensor, in full resolution. As previously noted, some loss of information from the original luminosity record may result, but it is typically at a rate that is imperceptible to the human eye. It is possible to have near lossless compression/decompression but, in the preferred embodiment, some controlled loss is accepted in order to further optimize the process (e.g., avoid adding precision that would not result in better image quality, as perceived by the user). As indicated by process block 440, conventional image processing (e.g., process steps 315-319) can now be applied to the uncompressed luminosity record for generating a color image for storage in a desired (e.g., typically standardized) file format. The end result is that one still ends up with a color digital image stored in one of the standardized formats (e.g., JPEG image 450). After processing the image, the PhotoServer may conveniently be used to further propagate the image, such as making the image available over the Internet to a multitude of other users (e.g., family and friends of the camera user). The image may, of course, also be rendered in hard copy, using a printing device available to the computer.

Note that, in accordance with the teachings of the present invention, the compute-intensive elements of digital image processing have been deferred such that they need not be performed at the digital camera but, instead, are deferred until the image arrives at the target platform (e.g., more powerful desktop or server computer). In this manner, the

amount of processing capability required at the camera is decreased, or for a given level of processing power the images may be processed faster at the digital camera. Additionally, the bandwidth required to transmit images from the digital camera to the target platform is substantially reduced, thereby facilitating wireless transmission of the images. At the same time, the present invention accomplishes this without losing the advantages of high compression, such as decreased storage requirement and decreased transmission time for downloading images.

Methodology and File Format for Wireless Progressive Transmission of Digital Images

A. Overview

The digital imaging device of the present invention incorporates a progressive transmission methodology allowing a user to capture a moment at a high level of quality, yet transmit a version of that image in a manner that is adapted for the communication bandwidth or medium currently available. The approach adopted by the present invention is to incorporate a software-implemented methodology (and supporting file format) into the imaging device itself that records (or is capable of recording) a moment in the full digital quality (desired by the user) but then prioritizes the image information into a variety of logical partitions or layers. Using this layered image information, the system, under control of a user, can choose how much or how little image information is actually transmitted to a target destination at a given point in time. Notwithstanding this flexibility, the user has at all times a full quality recording of the image.

Consider an embodiment of the methodology utilizing, for instance, nine layers. (The approach itself may be extended to any arbitrary number of layers or compartments, however). In such an embodiment, the first layer stores just enough image information to reproduce the image with a reasonable level of quality/resolution (e.g., suitable for Web page viewing). The second layer, containing additive image information, allows the image to be reproduced at a higher level of quality (and/or resolution), when combined with image information from the first layer. The ninth (i.e., final) layer contains the image information that, together with the eight earlier layers, helps reproduce the image at the highest quality/resolution levels desired.

In typical usage, therefore, the user records an image at its full quality on the digital film of his or her imaging device (or highest quality desired by the user), that is, utilizing all layers (e.g., the full nine-layer file for a nine-layer file embodiment). When uploading the image, however, the system optimizes transmission to only that layer or layers suitable for the then-existing communication medium, thereby reducing, if needed, the total amount of image information required to be sent and thus reducing transmission time and bandwidth requirements. Further in accordance with the present invention, the user may elect

to collaborate or synchronize the differences between the image information captured (e.g., full nine-layer file) with the image information already transmitted to a target destination (e.g., first layer only), at the user's convenience, so that the transmitted image may be upgraded to its final, full quality/resolution representation. In this manner, the present invention allows the user to record an image at its full digital quality/resolution yet send only so much of that image that is cost-effective and quick to transmit, at a given instance in time, based on the then-existing communication medium available.

B. Implementation as layer over existing compression

1. General

A number of techniques exist for compressing image information, including some of which may be viewed as de facto standards. Examples include, for instance, the aforementioned Discrete Cosine Transformation or DCT-based and wavelet transform-based compression methodologies. Rather than abandoning these techniques, the present invention is preferably embodied in a manner that accommodates them. This allows one to avoid "reinventing" image compression, which may itself cause new problems. Therefore, the present invention is preferably embodied as a functional layer on top of existing image compression. At the same time, however, the present invention is largely independent of the underlying image compression that is employed.

2. Embedded or layered coding

In image compression, embedded or layered coding refers to a compression technique where the image data is coded in multiple data sections, called "layers". The base layer represents the image at a particular resolution and quality. The "resolution" specifies how many pixels are employed. The "quality" specifies the signal/noise ratio, which is related to the degree of quantization that has occurred for the data. The interplay between these two may be represented graphically, as shown by the graph in Fig. 5A. Various combinations of resolution and quality may be selected for rendering a given image. Selection of combinations is simplified if the graph is viewed as being divided into various

layers, such as shown by the matrix or grid 550 in Fig. 5B. As shown, the two dimensions that affect each layer are resolution and quality. With each increasing layer, one adds to the resolution and/or adds to the quality. Thus, the layers cumulatively improve the resolution or quality (or both) of the image.

5 Examples of layered coding techniques include the JPEG progressive scanning method described by W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*, Chapman and Hall, New York, 1993, particularly at Chapter 6: *JPEG modes of operation*, and include the sub-band layered coding techniques described by D.Taubman and A. Zakhor, *Multirate 3-D subband coding of video*, IEEE Transactions on Image Processing, Vol. 3, No.5., September 1994, pages 572-588. The disclosures of each of the foregoing are hereby incorporated by reference. As commonly used, the term “sub-band coding” refers to the general class of image compression techniques (including the wavelet based coding method described earlier) where an image is first decomposed into various bands and then the bands are compressed using techniques suited to the band characteristics. The layered coding technique is perhaps best illustrated by way of example.

10
15
20 Consider the layered (progressive) coding technique supported in the JPEG standard, for example. The image is first segmented into 8x8 blocks, and the blocks are then transformed with the Discrete Cosine Transform (DCT) to result in 64 coefficients (positions 0 through 63) per block. These coefficients are quantized based on 64 quantized step sizes, prescribed one for each of the 64 positions. The base layer (i.e., Layer 00) may store only the DC band, that is, all of the coefficients at position zero. Continuing in this manner, the next layer stores coefficients 1 through 20, for instance. The next layer stores the remaining coefficients, 21 through 64. An alternative way to add more layers to this scheme is to divide up how the information is transmitted. For instance, the first layer may be transmitted followed by the most significant bits (or bit plane) of the next layer (i.e., to a certain bit depth). Note that the individual bitplanes are coded using entropy coding techniques.

25
For a wavelet-compressed image, for example, suppose that one has three resolution levels (e.g., as shown in Fig. 3B). Assume a uniform bit-depth of 8 bits for all levels after quantization. (In general, though, the quantization step size can be varied for

each band, leading to a different number of significant bitplanes in each band. The quantization step sizes are then transmitted with the image header). Assume bitplane 1 is the most significant bitplane and bitplane 8 is the least significant bitplane. Also assume that the original image is 640 pixels wide and 480 pixels high. The three resolution levels would then correspond to 160x120, 320x240, and 640x480, where WxH represents an image width of W pixels and height of H lines. Suppose that one wants to code the image with the layering scheme shown in Fig. 5B. The lowest resolution level (160x120) has a single band (LL) and the other resolution levels have 3 (LH, HL, HH) bands.

The base layer (i.e., Layer 00) may correspond to bitplanes 1 through 4 of the lowest resolution (160x120) level, Layer 10 may correspond to bitplanes 5 and 6 of the lowest resolution level, and Layer 20 may correspond to the remaining two bitplanes (7 and 8).

Layer 01 may store bitplanes 1 through 3 of the the 3 bands corresponding to the next higher resolution level (320x240), Layer 11 may store bitplanes 4 through 5, and layer 21 may store the remaing bitplanes (6 though 8) for that resolution level. Finally Layer 02 may correspond to bitplanes 1 through 2 of the three bands at the highest resolution level (640x480), followed by bitplanes 3 through 5 for Layer 12 and 6 through 8 for Layer 22. As can be seen, more bitplanes “at the same resolution level” are added along the quality axis, thereby improving the image quality at a fixed resolution. Similarly more bitplanes at the “next higher resolution level” are added along the resolution axis, thereby improving the image resolution.

The bitplanes in the different layers may be coded independently, or some dependencies may exist on preceding layers, as desired. More important is that the layers cumulatively add more information. Where dependencies exist, it is meaningless to transmit a layer that is dependent on layers that are not yet transmitted. Where dependencies do not exist, it is still useful to transmit the layers in a sequence that improves the picture quality and resolution. For example, in a layered coding scheme using JPEG progressive scanning, the first layer to be transmitted may be the DC coefficients that shows a crude representation of the image, the second layer transmitted may be AC coefficients 1 through 30 which further refines the image, finally followed by a third layer that codes the remaining coefficients.

C. Design considerations/assumptions

The following methodology is particularly suitable in instances where the main data storage (i.e., storage media) does not exist on the imaging device and the imaging device is typically untethered from the data storage. However, opportunities occur from time to time to transfer information to the main data storage, when the imaging device is occasionally tethered (e.g., to the data storage or to another device in communication with the data storage). Often in such a situation, other constraints exist, including time constraints and bandwidth constraints, such that only a small amount of data may be exchanged at a given instance in time. Against this backdrop, the user desires to get as much of his or her picture rendered as quickly as possible, thereby providing a good representation of the picture at the data storage side and perhaps even freeing up some storage at the imaging device.

As described above, digital image representation of a picture may be considered along two axes, resolution and quality (for a particular color depth). This presents the user with a matrix of choices. For example, the user may select a low-resolution image of 320 x 240 pixels in a camera capable of capturing 640 x 480 pixels, or the user may select a “high-quality” JPEG or a “low-quality” JPEG. In accordance with the present invention, the image information is layered so that the smallest-resolution, lowest-quality image is the lowest layer in the matrix (i.e., Layer 00). Each layer represents the smallest segment of information that is meaningful to send. Among the layers, Layer 00 represents the least amount of information needed for rendering a meaningful image. As one keeps adding layers along the axis of quality, more image data is added and the rendered picture becomes progressively higher in quality. Here, as one moves along the axis of quality, the image data becomes progressively less quantized as more and more bitplanes are transmitted. In a similar manner, one may progress along the axis of resolution, for obtaining a rendered image of increasingly higher resolution.

As the information between layers is somewhat dependent (in the format employed for the currently-preferred embodiment), transmission occurs in a manner such that the layers transmitted form a contiguous block of layers. There is no immediate use for transmitting information from a layer that cannot be interpreted or processed correctly until

receipt of another layer. As shown in Fig. 5C, for instance, this may be viewed as layers falling within a bounding box. Thus, the information transmitted is preferably within a rectangle that encloses the smallest region. Examples include, for instance, rectangle 561 and 563, each rectangle including layers that are contiguous.

Recall that each layer represents an atomic unit: the lowest segment of the data that can be sent. Each time the imaging device or camera is connected (directly or indirectly) to the main data storage, there exists an opportunity to send one or more extra layers. The actual constraints present are dependent on the user's given situation, such as whether the imaging device is connected via wireless or wireline communication. Therefore, how the layers are transmitted is, in a preferred embodiment, configurable (automatically and/or user-configurable) for a given communication scenario. In a wireline scenario, bandwidth is less expensive and thus more layers can be transmitted at a given instance in time. Here, the configuration may be controlled by the target device (i.e., server computer having the main data storage), so that the target device controls which layers it uploads. In a wireless scenario, on the other hand, bandwidth is more constrained. In that case, typically only the first layer would be uploaded. Additional layers would be uploaded at a later point in time (i.e., "updates"), either via a wireless or wireline communication medium. For the logical partitions shown in Fig. 5C, a maximum of nine communication sessions are required for transmitting the image at full resolution and full quality.

D. Data organization and processing

1. Preferred embodiment

Fig. 6A illustrates the basic data organization or progressive file format employed in a preferred embodiment of the present invention. Accompanying Fig. 6B represents the corresponding methodology 650 employed for organizing raw image data into the progressive file format 600, shown in Fig. 6A. Initially the picture is captured from the sensor (e.g., CMOS sensor), as represented by step 651. The raw image data represents a mosaic, arranged as repeating Bayer patterns, in memory (frame buffer). The mosaic is separated into a three separate planes, based on the color space employed, such as YUV or

RGB color space. This is indicated by step 652. In the preferred embodiment, YUV color space is employed. Therefore, the raw image data is separated into separate Y, U, and V planes. At this point, each plane is transformed (e.g., wavelet or DCT transformation) into separate quadrants or bands, as shown by step 653, and then quantized as shown by step 654.

5 For wavelet-transformed data, therefore, the image information has been decomposed into separate bands or quadrants (each to a certain bit depth), which have been quantized. Each bitplane within each band is coded (e.g., entropy coding), for compressing the data, as indicated by step 655. Within each bitplane, multiple passes of coding may be applied.

10 Although the information is coded, the length and location of each piece of data is preserved, so that the information may be organized into different layers, that is, inserted into different layers, as shown by the file format 600.

15 Now, the transformed, compressed and quantized image information is further organized into a quality/resolution matrix, using the various layers available from the file format 600 for Fig. 6A. Each layer is represented by two coordinates: resolution and quality. Organizing the image information into different layers includes selecting particular bands for a given layer, each band being represented to a particular bit depth. Layer 00 represents the lowest resolution/quality for the image; it will store the “smallest band.” Therefore, at this point, the system decides what will be the smallest layer, as indicated by step 656. This includes determining which bands to take for the image, and to what bit depth. For example, Layer 00 may contain bands 1 through 10, yet only include three bitplanes in band 1, two bitplanes in band three, and so forth and so on. This information is stored in Layer 00. Thus, Layer 00 itself is further subdivided, based on the bitplanes its stores. The storage methodology 650 continues in this manner, filling out the other layers (i.e., proceeding along quality/higher resolution) until the quality/resolution matrix is complete. Each layer 20 preferably includes information from all three separate color planes.

25 To assist with processing, the file format 600 stores housekeeping information in a header section. Specifically, housekeeping or attribute information for the picture is stored in the picture header 610, as shown in Fig. 6A. The picture header 610 includes capture-specific information or attributes about the picture, including width and height,

aperture and exposure time used to capture the picture, analog gains of the (CMOS) sensor, timestamp, and the like. The header also includes information on the quantization step sizes employed for the various bands in the image. If desired, the picture header 610 may include a thumbnail image (bitmap), which is useful for providing a quick preview of the image. The thumbnail could be compressed using any coding technique.

Additionally, the file format 600 stores housekeeping information about the layers, using a layer directory 620. Each layer itself is stored, in a preferred embodiment, as a separate record, such as a separate physical file in a flash file system. The layer directory 620 includes an entry for each layer record; the entry includes the name of the accompanying physical file (text string) and length of the file (numeric value). Each record, such as record 630, includes a record header (e.g., record header 631) for storing housekeeping information about that particular record. For a YUV color space embodiment, for instance, the record header 631 stores Y offset and length, U offset and length, and V offset and length. This is followed by the record body 635, which stores the image information proper. For a YUV color space embodiment, this would include, for example, separate Y data, U data, and V data compartments or sections, within the record.

The methodology 700 of the present invention for transferring image data is illustrated in Fig. 7. When the imaging device is first connected to the target device, the header information is transmitted, so that the target device includes the basic information about the picture (e.g., including a thumbnail). This is illustrated in step 701. Next, at step 702, the target device reads the record directory of an image to control uploading of that image. Now, the target device (e.g., server computer) is ready to access and fetch the individual layers that comprise the image information. Starting with the first record entry, the target device successively fetches the corresponding records (referenced by the recorded directory entries), as illustrated by step 703. As each record is retrieved, the server computer may reset the record entry (e.g., setting the filename to NULL) on the imaging device-side, for indicating that that record has already been fetched. Once the entry has been set to NULL, the corresponding record may be deleted. At the same time, at the server computer, the corresponding record entry is updated to now reference the just-fetched record. This

allows the imaging device to know which records have been sent, and allows the server computer to know which records have been received. Upon transmission of the layer, the communication connection may be (optionally) now closed, as indicated by step 704.

5 The server computer fetches successive record(s) at the next opportunity, either during the current connection or during the next connection (indicates that the current connection is now terminated). Step 705 illustrates that a loop is established to jump back to step 703 for transmitting the next (remaining) layer. The server computer itself may be configured to fetch the layers in a particular order, such as first increasing the quality before increasing the resolution, or conversely increasing the resolution before increasing the quality. As previously described, however, the server computer should preferably retrieve layers such that a bounding box encloses all retrieved layers. Once no more layers remain to be transmitted (a fact which is uncovered at step 705), the method is done.

10 As an example of real-life application of the foregoing methodology consider, for instance, a user "in the field" who needs to capture an image and transmit it back to his or her office. Here, the user may initially be interested in only a low quality version (say, the first layer only) of the image for transmission back to the office. The device though records a high quality/resolution version of the image. There is no loss of image information in the high quality/resolution version of the image if no quantization is employed (i.e., the device then would store a lossless version of the image), but typically small quantization step sizes are employed so as to control the size of the high quality/resolution version of the image.

15 Only the first layer is transmitted back to a target device at the user's office (e.g., a desktop or server computer with Internet connectivity). At a later point in time, for instance when time and/or bandwidth are not an issue, the user can transmit all remaining layers for the image.

20 Here, the remaining layers need not be directly transmitted to the destination device. Instead, the user can, for example, transmit the remaining layers to yet another device, such as his or her home computer, which then may synchronize or collaborate the information with that already present at the destination device, for generating a copy of that image at full quality (at either one or both of the devices). As demonstrated by this example, the present invention allows the user to avoid the classic tradeoff of image quality versus transmission time. In

particular, the user need not endure long transmission time in order to transmit a copy of the image, nor is the user forced to only record a low quality version of the image for transmitting.

5

2. Alternative embodiments

Those skilled in the art will appreciate that the foregoing data organization may be modified to accommodate the peculiarities or constraints of a given environment. For instance, the foregoing data organization may also be embodied in devices that do not readily support record-oriented organization. In such a case, the image information is simply organized into a single blob (binary large object), which may be subdivided into separate logical sections corresponding to the previously-mentioned header and record sections. Also, the directory entries are modified to store offset/length entries (instead of filenames for physical files). Fig. 8 illustrates the basic approach. Fig. 8 demonstrates a recordless file format 800 in that it does not provide individual records (i.e., named files) per se. Instead, the information is organized such that the highest priority layer occurs at the bottom of the blob object and the lowest priority layer occurs at the top. The server accesses the structure by taking increasing amounts of the blob, so that the corresponding image file at the server becomes increasingly larger (as additional logical layers are uploaded). As each layer is uploaded to the server, the corresponding memory at the imaging device may be freed, so that it can be used for other purposes. Thus as shown, there is no need for the records to exist as separate physical files.

10

20

20

E. Conclusion

In accordance with the present invention, image information after wavelet decomposition and quantization is sorted into successive layers or compartments, based on quality and resolution criteria. No new image information is lost during the layering process, nor has any numeric operation (e.g., floating point operation) has been performed which would render the information less accurate. Instead, the image information is organized in a fashion that allows the system to control the transmission of varying qualities/resolutions of a

25

given image from the progressive file format. Since the user is able to capture an image at full quality, yet choose what quality/resolution the image is transmitted at, the user is not forced to capture poor quality images just to have good transmission times. Instead, the user can enjoy high-quality images as well as good transmission times.

5

10

While the invention is described in some detail with specific reference to a single-preferred embodiment and certain alternatives, there is no intent to limit the invention to that particular embodiment or those specific alternatives. For example, those skilled in the art will appreciate that the progressive transmission methodology may be readily adapted to accommodate scenarios where a full-quality image is not always recorded at the device, such as in instances where the user knows beforehand that only a particular resolution will ever be needed. Thus, the true scope of the present invention is not limited to any one of the foregoing exemplary embodiments but is instead defined by the appended claims.

100-00000000